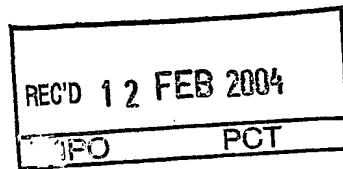


BUNDE REPUBLIK DEUTSCHLAND

EP/03/12098

19. 11. 2003



**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen: 103 37 825.1

Anmeldetag: 18. August 2003

Anmelder/Inhaber: Siemens Aktiengesellschaft, München/DE

Bezeichnung: Verfahren zur Erzeugung eines Bitstroms
aus einem Indizierungsbaum

Priorität: 15.11.2002 DE 102 53 275.3

IPC: H 04 N 7/24

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 10. November 2003
Deutsches Patent- und Markenamt

Der Präsident
Im Auftrag

Schmidt C.

Beschreibung

Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum

5

Die vorliegende Erfindung betrifft ein Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum sowie ein Codierverfahren und ein Decodierverfahren, welche von dem Verfahren zur Erzeugung des Bitstroms Gebrauch machen. Ferner

10 betrifft die Erfindung eine Codier- und Decodierzvorrichtung.

Es ist häufig erforderlich, bestimmte Inhalte aus einem Bitstrom entsprechend einer von einem Benutzer vorab getroffenen Abfrage auszulesen bzw. hinsichtlich bestimmter Inhalte zu 15 ermitteln, ob diese Inhalte in dem Bitstrom überhaupt enthalten sind. Eine von einem Benutzer definierte Abfrage kann hierbei mittels einer Abfragesprache, wie SQL (siehe Referenz [1]) oder XPATH (siehe Referenz [2]) formuliert werden. Es ist hierbei vorteilhaft, wenn nicht der gesamte Bitstrom nach 20 den gewünschten Inhalten durchsucht werden muss, sondern wenn die Informationen in einer Indizierungsliste oder in einem Indizierungsbaum abgelegt sind, so dass lediglich der Teil des Bitstroms, in dem der Indizierungsbaum bzw. die Indizierungsliste gespeichert sind, durchsucht werden muss.

Das Problem des Auslesens von Daten aus einem Bitstrom ergibt sich beispielsweise bei einem mit Hilfe der XML-Sprache (XML = extensible markup language) erstellten Dokument, das im MPEG7-BiM Format repräsentiert wird. Bezüglich des MPEG7-BiM 30 Formats eines XML-Dokuments sei hierbei insbesondere auf das Dokument [3] verwiesen. Bei einer derartigen Repräsentation wird der erzeugte Bitstrom in mehrere Einheiten (access units) aufgeteilt, welche ihrerseits wiederum aus mehreren Fragmenten (fragment update units) bestehen. Die Einheiten 35 werden codiert und bei Bedarf als MPEG7-BiM-Strom an einen oder mehrere Empfänger versandt.

In Bezug auf die Abfrage von Informationen aus XML-Dokumenten sind bereits eine Vielzahl von Abfragesprachen bekannt, die es erlauben, nach bestimmten Informationen in dem Dokument zu suchen. Beispielhaft sei an dieser Stelle auf die bereits erwähnte Abfragesprache XPATH (siehe Referenz [2]) verwiesen. Mit der Abfragesprache XPATH können Auswahlkriterien zur Filterung gewünschter Informationen innerhalb eines XML-Dokuments definiert werden. Ziel einer Abfrage kann hierbei einerseits die Bewertung sein, ob eine Einheit des Bitstroms für den Empfänger wichtig ist. Andererseits kann mit Hilfe einer Abfrage gezielt auf gewünschte Informationen in dem XML-Dokument zurückgegriffen werden. Das MPEG7-Codierverfahren hat bis jetzt bei der Erzeugung des Bitstroms eines XML-Dokuments keine Mechanismen vorgesehen, die einen wahlfreien Zugriff auf bestimmte Elemente des XML-Dokuments ermöglichen. Der MPEG7-Bitstrom muss deshalb zur Suche nach Elementen decodiert werden. Man erhält dann wieder ein Dokument im XML-Format, welches mittels der Abfragesprache XPATH durchsucht werden kann. Das Decodieren und anschließende Abarbeiten eines XML-Dokuments zur Suche nach spezifischen Inhalten ist somit sehr zeitaufwendig und für bestimmte zeitkritische Anwendungen unakzeptabel. Ferner kann das Problem auftreten, dass der Speicher im Decoder begrenzt ist, so dass der Bitstrom nicht vollständig decodiert wird. Darüber hinaus war der Aufwand des Decodierens unnötig, falls die am decodierten XML-Dokument durchgeführte XPATH-Abfrage negativ verläuft.

Im Rahmen von TV-Anytime (TVA), das in Dokument [4] beschrieben ist, wird eine Indexstruktur verwendet, die einen wahlfreien Zugriff auf bestimmte Elemente eines Datenfragmentes erlaubt. Die Indexstruktur besteht aus mehreren Teilen und umfasst eine sog. Key-Index-Liste, in der sämtliche indizierten Pfade eines Dokumentes abgelegt werden. Bei einer Abfrage werden diese Pfade der Reihe nach mit der Abfrage verglichen, bis ein entsprechender Eintrag in der Key-Index-Liste gefunden wird. Durch die Informationen, die in der Key-Index-Liste

zu diesem Eintrag gespeichert sind, können die Stellen in einem Beschreibungsstrom bestimmt werden, an denen der indizierte Eintrag codiert vorliegt. Durch die Verwendung der Key-Index-Liste ist es nicht mehr erforderlich, uninteressante Datenfragmente zu decodieren, so dass weniger Speicher-

5 platz während einer Abfrage benötigt wird. Das lineare Durchwandern der Key-Index-Liste ist aber zeitkonsumierend und die Übertragung aller indizierter Pfade ist aufwendig.

10 Aufgabe der Erfindung ist es deshalb, ein Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum zu schaffen, bei dem die Suche nach indizierten Daten einfach und effizient ermöglicht wird.

15 Diese Aufgabe wird gemäß den Merkmalen der unabhängigen Patentansprüche gelöst. Weiterbildungen der Erfindung ergeben sich aus den abhängigen Ansprüchen.

20 Der Indizierungsbaum, aus dem mit dem erfindungsgemäßen Verfahren ein Bitstrom erzeugt wird, umfasst eine Mehrzahl von Hierarchieebenen, wobei jeder Hierarchieebene ein oder mehrere Indexknoten zugeordnet sind und die Indexknoten Indexdaten enthalten, die in dem Indizierungsbaum nach einem oder mehreren vorgegebenen Kriterien sortiert sind. Bei dem erfindungsgemäßen Verfahren werden die Indexdaten der Indexknoten in den Bitstrom eingefügt und ferner wird für einen Indexknoten jeweils die Information in den Bitstrom eingefügt, an welcher Stelle im Bitstrom sich die Indexdaten von einem oder mehreren Indexknoten der Hierarchieebene, die unterhalb der Hie-

30 rarchieebene des jeweiligen Knotens liegt, befinden. Durch die Speicherung der zusätzlichen Information bezüglich der Indexknoten in einer darunter liegenden Hierarchieebene wird eine Suche nach bestimmten Indexdaten wesentlich vereinfacht, da hierdurch Sprünge zu den für die Suche relevanten Index-

35 knoten ermöglicht werden. Somit wird eine wesentlich effizientere Abfrage und Suche nach Indexdaten gewährleistet.

In einer bevorzugten Ausführungsform ist der Indizierungsbaum als sog. B-Tree (= Balanced Tree) aufgebaut, der eine ausgewogene Verteilung der Daten auf die Knoten des Baums gewährleistet. Eine genaue Beschreibung des B-Trees findet sich 5 in der Referenz [5].

Die Indexdaten in dem Indizierungsbaum können nach beliebigen Kriterien sortiert sein, beispielsweise lexikographisch.

10 In einer besonders bevorzugten Ausführungsform sind die Indexdaten nach der Depth-First-Ordnung in den Bitstrom eingefügt. Durch die Verwendung der Depth-First-Ordnung werden die Indexdaten in dem Indizierungsbaum zunächst der Tiefe nach in den Bitstrom eingefügt, wodurch die für eine Abfrage relevanten Informationen im Bitstrom benachbart zueinander angeordnet werden und nicht relevante Informationen effizient übersprungen werden können. Eine genaue Beschreibung der Depth-First-Ordnung findet sich in der Referenz [6].

15 20 In einer weiteren bevorzugten Ausführungsform der Erfindung umfassen die Indexdaten Pfade eines Dokumentenstrukturbaums, der einen Wurzelknoten und eine Mehrzahl von Blattknoten umfasst. Ferner umfassen die Indexdaten vorzugsweise die Werteausprägungen der Pfade und die Positionen der Werteausprägungen in dem Dokument, das durch den Dokumentenstrukturbau dargestellt ist. Darüber hinaus umfassen die Indexdaten in einer bevorzugten Ausführungsform auch noch die Anzahl der Pfade in einem Indexknoten.

25 30 35 Die Pfade können hierbei Absolutpfade sein, die von dem Wurzelknoten des Dokumentenstrukturbaums beginnen und zu einem Blattknoten des Dokumentenstrukturbaums führen. In einer besonders bevorzugten Ausführungsform der Erfindung sind die Pfade jedoch relative Pfade, wobei ein relativer Pfad eines jeweiligen Indexknotens ein Pfad relativ zu einem zuvor in den Bitstrom eingefügten Pfad des jeweiligen Indexknotens oder eines Indexknotens einer über der Hierarchieebene des

jeweiligen Indexknotens liegenden Hierarchieebene ist. Durch die Verwendung von relativen Pfaden werden Gemeinsamkeiten in den Pfaden ausgenutzt, denn die Pfade von benachbarten Knoten haben meist einen gemeinsamen Anteil. Auf diese Weise kann

5 der benötigte Speicherbedarf für die Indexdaten im Bitstrom reduziert werden. Eine weitere Reduzierung des Speicherbedarfs kann dadurch erreicht werden, dass die Pfade des Indexknotens, dessen Indexdaten als erste einer Hierarchieebene in den Bitstrom eingefügt werden, in einer Reihenfolge umgekehrt 10 zu der Reihenfolge, in der die Indexdaten in dem Indexknoten angeordnet sind, in den Bitstrom eingefügt werden. Hierbei wird die Tatsache berücksichtigt, dass die Indexdaten am Ende des ersten Indexknotens einer Hierarchieebene eine größere Ähnlichkeit zu dem Indexknoten der nächsthöheren Hierarchie- 15 ebene aufweisen als die Indexdaten am Anfang des ersten Indexknotens. Folglich ergibt sich in bestimmten Fällen eine besonders effektive Codierung mittels von relativen Pfaden.

20 In einer besonders bevorzugten Ausführungsform der Erfindung umfassen die Pfade Beschreibungselemente eines XML-Dokuments (XML = extensible markup language), wobei die Pfade insbesondere XPATH-Pfade des XML-Dokuments sind.

In einer weiteren Ausführungsform der Erfindung sind die Indexdaten mit einem Codierverfahren binär codiert, insbesondere mit einem MPEG-Codierverfahren. In einer besonders bevorzugten Ausführungsform wird als Codierverfahren das MPEG7-Codierverfahren verwendet.

30 Neben dem oben beschriebenen Verfahren zur Erzeugung eines Bitstroms umfasst die Erfindung ferner ein Verfahren zum Codieren einer Datenstruktur, bei dem die Datenelemente der Datenstruktur in einem Indizierungsbaum indiziert sind, wobei ein Bitstrom gemäß dem erfindungsgemäßen Verfahren erzeugt 35 wird und dieser Bitstrom Teil des codierten Datenstroms ist. Darüber hinaus umfasst die Erfindung ein Verfahren zum Decodieren einer Datenstruktur, wobei das Verfahren derart aus-

gestaltet ist, dass eine mit dem soeben beschriebenen Codierverfahren codierte Datenstruktur decodiert wird.

5 Darüber hinaus umfasst die Erfindung ein Verfahren zum Codieren und Decodieren einer Datenstruktur, welches das oben beschriebene Codierverfahren und Decodierverfahren umfasst.

10 Die Erfindung umfasst auch eine Codiervorrichtung, mit der das erfindungsgemäße Codierverfahren durchführbar ist, sowie eine Decodiervorrichtung, mit der das erfindungsgemäße Decodierverfahren durchführbar ist. Ferner betrifft die Erfindung eine entsprechende Vorrichtung zum Codieren und Decodieren, mit der das oben beschriebene kombinierte Codier- und Decodierverfahren durchführbar ist.

15 15 Ausführungsbeispiele der Erfindung werden nachfolgend anhand der Zeichnungen dargestellt und erläutert.

20 Es zeigen

Fig.1 ein Beispiel eines in dem erfindungsgemäßen Verfahren verwendeten Indizierungsbaums;

Fig.2 einen Ausschnitt aus dem in Fig. 1 gezeigten Indizierungsbaum;

Fig.3 einen Ausschnitt aus einem Bitstrom, der aus dem Indizierungsbaum der Fig. 1 mit dem erfindungsgemäßen Verfahren generiert wurde;

30 Fig.4 einen vereinfachten Indizierungsbaum, in dem die Indexdaten lexikographisch geordnet sind und Pfade eines Dokumentenstrukturbaums darstellen;

35 Fig.5 den Indizierungsbaum der Fig. 4, wobei die Pfade der zweiten Hierarchieebene teilweise als relative Pfade dargestellt sind;

Fig.6 den Indizierungsbaum der Fig. 4, wobei alle Pfade der zweiten Hierarchieebene als relative Pfade dargestellt sind; und

5

Fig.7 eine Abwandlung des Indizierungsbaums gemäß Fig. 4.

Figur 1 zeigt beispielhaft einen in dem erfindungsgemäßen Verfahren verwendeten Indizierungsbaum. Es handelt sich bei 10 diesem Baum um einen sog. B-Baum (B steht für Balanced), welcher häufig bei der Indizierung von Daten zur Anwendung kommt. Die Struktur eines B-Baums ist beispielsweise in der bereits erwähnten Referenz [5] beschrieben. Der B-Baum weist eine ausgeglichene Struktur auf und ermöglicht hierdurch ein 15 Suchen mit einem gemessen an der Zahl der Einträge nur logarithmischen Aufwand, wohingegen beim Suchen mit einer Liste ein linearer Aufwand erforderlich ist.

Der Indizierungsbaum enthält eine Mehrzahl von Knoten 1 bis 20 10, wobei in jedem Knoten eine oder mehrere Indexdaten in Form von sog. Schlüsseln gespeichert sind. Im vorliegenden Fall sind die Indexdaten sog. XPATH-Pfade eines Beschreibungsbaums eines XML-Dokuments. Eine Beschreibung von XPATH-Pfaden findet sich in der bereits erwähnten Referenz [2]. Die XPATH-Pfade führen von dem Wurzelknoten des Beschreibungsbaums des XML-Dokuments zu den einzelnen Blattknoten des Beschreibungsbaums des XML-Dokuments. Es werden in dem in dieser Ausführungsform verwendeten Beschreibungsbaum als Blattknoten XML-Elemente mit einfachem Inhalt und alle XML- 30 Attribute verwendet. Folglich umfassen die Indexdaten des in Figur 1 gezeigten Indizierungsbaums Wurzel-Blatt-Pfade eines XML-Beschreibungsbaums.

Die XPATH-Pfade sind in dem Indizierungsbaum der Fig. 1 als 35 eine Reihe aus Groß- und Kleinbuchstaben dargestellt und lexicographisch geordnet, wobei jeder Buchstabe für einen Elementnamen in dem XML-Beschreibungsbaum steht. Beispielsweise

bedeutet der Pfad MDMVUFACN in Knoten 1 MPEG7/Description/MultimediaContent/Video/UsageInformation/FinancialResults/AccountItem/Costtype/Name. Neben dem Pfad sind in den Knoten auch die Werteausprägungen gespeichert, in denen dieser Pfad 5 in dem XML-Dokument auftritt. Im vorliegenden Fall tritt der Pfad MDMVUFACN im Indexknoten 1 zweimal mit den Ausprägungen "Total for Production" und "Broadcast" auf. Demgegenüber tritt beispielsweise der Pfad MDMVUAFAC im Indexknoten 2 nur einmal mit der Ausprägung "EUR" auf.

10

Die Nummerierung der Indexknoten in dem Indizierungsbaum der Figur 1 entspricht der Reihenfolge, in welcher die Indexdaten mit Hilfe des erfindungsgemäßen Verfahrens in den Bitstrom eingefügt werden. Die Reihenfolge entspricht hierbei einer 15 sog. Depth-First-Ordnung, die beispielsweise in der bereits erwähnten Referenz [6] beschrieben ist. Durch das Einfügen der Indexdaten in den Bitstrom gemäß der Depth-First-Ordnung ergibt sich eine geringere Komplexität beim Zugriff auf die Indexdaten, da bei der Suche nach Indexdaten nicht relevante 20 Informationen übersprungen werden.

In Figur 2 ist ein Ausschnitt aus dem Indizierungsbaum der Figur 1 gezeigt, an dem an einem Beispiel die Abfrage von Indexdaten in diesem Baum erläutert wird. Es handelt sich hierbei um die Abfrage, ob der XPATH-Pfad MPEG7/Description/Multimediacontent/Video/UsageInformation/Availability/Dissemination/Disseminator/Agent/Name die Werteausprägung "Discovery" enthält (in Kurzschreibweise MDMVUADDAN = "Discovery"). Da der Suchstring MDMVUADDAN lexikographisch kleiner 30 ist als der XPATH-Pfad "MDMVUFACN" des Indexknotens 1 wird auf den Indexknoten 2 zurückgegriffen. Der Suchstring ist wiederum lexikographisch kleiner als der Eintrag "MDMVUAFAC" des Indexknotens 2. Deshalb wird auf den links des Indexknotens 2 gelegenen Indexknoten 3 übergegangen. Dieser Indexknoten 35 enthält zwei XPATH-Pfade "MDMVUAAAt" und "MDMVUADFh". Der Suchstring liegt zwischen diesen beiden XPATH-Pfaden, so dass in der nächsten Hierarchieebene auf den zwischen Indexknoten

4 und Indexknoten 6 liegenden Indexknoten 5 übergegangen wird. Dieser Indexknoten enthält nunmehr als erstes Element den gesuchten XPATH-Pfad "MDMVUADDAN" sowie die gesuchte Werteausprägung "Discovery". Es wurde somit in dem Indexknoten 5 die passende Werteausprägung gefunden und man erhält dann die Position des entsprechenden Elementes im indizierten XML-Dokument. Dadurch, dass die Indexknoten in den Bitstrom gemäß der Depth-First-Ordnung eingefügt sind, werden alle weiteren Knoten des Indizierungsbaums bei der Suche ignoriert bzw. übersprungen, so dass die Suche nach Elementen im Indizierungsbaum geringe Komplexität aufweist.

Im folgenden wird detailliert auf die Einfügung der Indexdaten in den Bitstroms mittels des erfindungsgemäßen Verfahrens eingegangen. Dies geschieht anhand der Figur 3, welche einen Ausschnitt aus einem mit dem erfindungsgemäßen Verfahren erzeugten Bitstrom darstellt. In dem in Figur 3 gezeigten Bitstrom sind übersichtshalber die Nummern der Indexknoten aufgeführt, wobei diese Nummern jedoch nicht mit dem Strom übereinmittelt werden. Im Detail sind in Figur 3 die Inhalte der Indexknoten 1 und 3 gezeigt. Die Knoten enthalten als erstes Element zunächst die Anzahl der Einträge, d.h. die Anzahl der in dem Knoten gespeicherten XPATH-Pfade. Für den Indexknoten 1 ergibt sich hier die Zahl 1 und für den Indexknoten 3 die Zahl 2. Anschließend folgt für jeden Eintrag der Schlüssel, d.h. der jeweilige XPATH-Pfad sowie die Anzahl der Instanziierungen, d.h. die Anzahl der Werteausprägungen, die dieser XPATH-Pfad in dem XML-Dokument hat. Für den Indexknoten 1 ergibt sich beispielsweise für den XPATH-Pfad "MDMVUAFCN" die Instanzierungsanzahl 2, da der Pfad mit den beiden Werteausprägungen "Total for Production" und "Broadcast" in dem XML-Dokument auftritt. Für jede Werteausprägung wird ferner die Position dieser Werteausprägung in dem XML-Dokument in den Bitstrom eingefügt. Es ist bei dem erfindungsgemäßen Verfahren wesentlich, dass neben diesen Informationen ferner in den Bitstrom die Information eingefügt wird, an welcher Position im Bitstrom - insbesondere an welcher Position relativ zur

momentanen Position (offset) - sich die Indexdaten von mit dem jeweiligen Indexknoten verbundenen Indexknoten der darunter liegenden Hierarchieebene befinden. Hierdurch wird das Überspringen von Knoten ermöglicht, die für eine vorgegebene 5 Suchabfrage nicht relevant sind.

Im folgenden bezeichnet man die sich von einem Knoten in eine tiefer liegende Hierarchieebene verzweigenden Knoten als Kindknoten. Es ist bei dem erfindungsgemäßen Verfahren zu beachten, dass die Information bezüglich der Position des ersten Kindknotens einer Hierarchieebene nicht gespeichert werden muss, da dieser Kindknoten im Bitstrom als nächster eingelesen wird. Wie der Figur 3 entnommen werden kann, werden somit für den Indexknoten 1 lediglich die Position des Indexknotens 10 (als "offset(10)" bezeichnet) und für den Indexknoten 3 lediglich die Positionen der Indexknoten 5 und 6 (bezeichnet als "offset(5)" und "offset(6)") gespeichert. Ergibt sich beispielsweise, dass die gesuchte Information bei einer Abfrage lexikographisch größer als der XPATH in Knoten 1 ist, wird sofort zum Knoten 10 gesprungen, so dass es nicht mehr notwendig ist, den Indexknoten 2 und die sich von dem Indexknoten 2 verzweigenden Indexknoten zu durchsuchen. Hierdurch wird eine effektive Suche ermöglicht. 10 15 20

In der nachfolgenden Tabelle ist nochmals aufgeführt, welche Informationen pro Knoten in dem Bitstrom gespeichert werden.

Tabelle:

Knoten {
Anzahl der Einträge
für jeden Eintrag:
Schlüssel
Anzahl der Instanziierungen
für jede Instanziierung:
Werteausprägung
Position
für jeden Kindknoten außer den ersten
Offset im Strom

Der mit dem erfindungsgemäßen Verfahren erzeugte Bitstrom wird an den Empfänger übermittelt und kann entsprechend einer Abfrage, die beispielhaft in Bezug auf Figur 2 erläutert wurde, ausgelesen werden. Durch Vergleiche zwischen dem abzufragenden XPATH und den Einträgen in den Indexknoten werden so lange bestimmte Knoteninformationen aus dem Strom ausgelesen, bis der entsprechende gesuchte XPATH mit der gesuchten Werteausprägung gefunden wurde oder bis kein passender XPATH-Pfad aufgrund der Sortierung mehr möglich ist. Im letzteren Fall ist die gewünschte Information im indizierten Dokument nicht enthalten.

15

Im folgenden werden anhand der Figuren 4 bis 7, welche vereinfachte Indizierungsbäume mit lexikographisch geordneten Einträgen darstellen, weitere Ausführungsformen des erfindungsgemäßen Verfahrens erläutert. In Figur 4 ist ein mittels des erfindungsgemäßen Verfahrens zu codierender Indizierungsbäum gezeigt, der Knoten K1 bis K4 umfasst. Der Knoten K1 wird hierbei als Vaterknoten bezeichnet und die sich aus dem Knoten K1 verzweigenden Knoten K2 bis K4 werden als Kindkno-

ten bezeichnet. Bei den Einträgen handelt es sich um Pfade eines Dokumentenstrukturbauums. Statt im Bitstrom die Pfade mit ihrer gesamten Länge zu übertragen, besteht die Möglichkeit, sog. relative Pfade in dem zu übertragenden Indexknoten abzuspeichern, wobei diese Pfade relativ in Bezug auf einen im vorangegangenen übertragenen Pfad des Vaterknotens bzw. in Bezug auf einen im vorangegangenen übertragenen Pfad des momentan zu übertragenden Indexknotens sind.

- 5 10 Die Einfügung von relativen Pfaden in den Bitstrom ist in Figur 5 verdeutlicht. Hierbei bedeutet die Darstellung "..." einen Schritt zurück in dem Pfad, auf den sich der relative Pfad bezieht. In dem Knoten K1 wird für den Eintrag "AE" der relative Pfad "...E" in Bezug auf den vorangegangenen Eintrag 15 ACB übertragen. In den Kindknoten K1, K2 und K3 werden jeweils in den zweiten und dritten Einträgen die relativen Pfade in Bezug auf den ersten Pfad des jeweiligen Kindknotens übertragen.
- 20 Bei der Ausführungsform der Figur 5 wird für die ersten Pfade AA, ACC bzw. AEF der Kindknoten K2, K3 bzw. K4 der komplette absolute Pfad übertragen. Es ist nun zusätzlich möglich, auch die ersten Pfade der Kindknoten als relative Pfade in Bezug auf einen Pfad des Vaterknotens K1 zu übertragen. Dies ist in Figur 6 dargestellt. Anstatt der Pfade AA und ACC der ersten und zweiten Kindknoten werden hierbei die relativen Pfade "...A" und ".C" in bezug auf den ersten Pfad des Vaterknotens übertragen. Anstatt des Pfads "AEF" des dritten Kindknotens wird der relative Pfad "F" in bezug auf den zweiten Pfad des 30 Vaterknotens übertragen. Die Pfade werden hierbei vorzugsweise in ihrer Bytedarstellung in den Bitstrom abgelegt.

Durch die Verwendung von relativen Pfaden wird das Datenvolumen im Strom stark verringert, da bei der Abspeicherung der 35 Pfade Gemeinsamkeiten in den Pfaden ausgenutzt werden, so dass weniger Speicherplatz benötigt wird. Insbesondere müssen

bei relativen Pfaden die jeweiligen gemeinsamen Anteile der Pfade nur einmal übermittelt werden.

In einer weiteren Ausführungsform der Erfindung wird eine Datenersparnis dadurch erreicht, dass bei der Anordnung der Einträge der Indexknoten für den ersten Kindknoten K2 eine gegenüber der Figur 4 umgekehrte Reihenfolge verwendet wird. Hierdurch wird erreicht, dass der Eintrag, der die größte Ähnlichkeit zu dem vorangegangenen Vaterknoten K1 aufweist, 5 am Anfang des Kindknotens K2 steht. Damit können sich bei speziellen Ausgestaltungen des Indizierungsbaums wesentlich verkürzte relative Pfade und ein verringelter Speicherbedarf 10 ergeben.

15 Die Pfade können die bereits im vorangegangenen erwähnten XPATH-Pfade eines XML-Beschreibungsbaums sein. Ferner können die Pfade binär codiert sein, beispielsweise mit einem MPEG7-Codierverfahren. Insbesondere können die in der bereits erwähnten Referenz [3] spezifizierten Verfahren zur Binärcodierung 20 eingesetzt werden. Der Einsatz von binär codierten Paden kann im Vergleich zu textuellen Repräsentationen zu Einsparungen bei der Anzahl der zu übermittelnden Bits führen.

Literaturverzeichnis:

- [1] <http://dx1.hrz.uni-dortmund.de:8001/doc1/hrz/sqlref/sqloracle.html>
- 5 [2] <http://www.w3.org/TR/xpath>
- [3] ISO/IEC 15938-1 Multimedia Content Description Interface
- Part 1: Systems, Geneva 2002
- [4] TV-Anytime Specification Series S-3 on Metadata, Part-B,
Version 13
- 10 [5] <http://www.public.asu.edu/~peterjn/btree/>
- [6] http://www.generation5.org/simple_search.shtml

Patentansprüche

1. Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum, wobei der Indizierungsbaum eine Mehrzahl von Hierarchieebenen umfasst und jeder Hierarchieebene ein oder mehrere Indexknoten (K1, K2, K3, K4) zugeordnet sind, wobei die Indexknoten (K1, K2, K3, K4) Indexdaten enthalten, die in dem Indizierungsbaum nach einem oder mehreren vorgegebenen Kriterien sortiert sind, bei dem:

10

- die Indexdaten der Indexknoten (K1, K2, K3, K4) in den Bitstrom eingefügt werden;
- für einen Indexknoten jeweils die Information in den Bitstrom eingefügt wird, an welcher Stelle im Bitstrom sich die Indexdaten von einem oder mehreren Indexknoten (K1, K2, K3, K4) der Hierarchieebene, die unterhalb der Hierarchieebene des jeweiligen Knotens liegt, befinden.

20

2. Verfahren nach Anspruch 1, bei dem der Indizierungsbaum ein B-Tree (= Balanced Tree) ist.

3. Verfahren nach Anspruch 1 oder 2, bei dem die Indexdaten in dem Indizierungsbaum lexikographisch sortiert sind.

4. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die Indexdaten nach der Depth-First-Ordnung in den Bitstrom eingefügt werden.

30

5. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die Indexdaten Pfade eines Dokumentenstrukturbaums bestehend aus mindestens einen Wurzelknoten und einer Mehrzahl von Blattknoten umfassen.

6. Verfahren nach Anspruch 5, bei dem die Indexdaten die Werteausprägungen der Pfade und die Positionen der Werte-

35

ausprägungen in dem Dokument, das durch den Dokumentenstrukturbau dargestellt ist, enthalten.

7. Verfahren nach Anspruch 5 oder 6, bei dem die Indexdaten
5 die Anzahl der Pfade in einem Indexknoten (K1, K2, K3, K4) umfassen.

8. Verfahren nach einem der Ansprüche 5 bis 7, bei dem die
10 Pfade absolute Pfade umfassen, die von dem Wurzelknoten
beginnen und zu einem Blattknoten führen.

9. Verfahren nach einem der Ansprüche 5 bis 8, bei dem die
15 Pfade relative Pfade umfassen, wobei ein relativer Pfad
eines jeweiligen Indexknotens (K1, K2, K3, K4) ein Pfad
relativ zu einem zuvor in den Bitstrom eingefügten Pfad
des jeweiligen Indexknotens (K1, K2, K3, K4) oder eines
Indexknotens (K1, K2, K3, K4) einer über der Hierarchie-
ebene des jeweiligen Indexknotens (K1, K2, K3, K4) lie-
genden Hierarchieebene ist.

20 10. Verfahren nach Anspruch 9, bei dem die Pfade des Index-
knotens, dessen Indexdaten als erste einer Hierarchiebe-
ne in den Bitstrom eingefügt werden, in einer Reihenfolge
umgekehrt zu der Reihenfolge, in der die Indexdaten in
dem Indexknoten (K1, K2, K3, K4) angeordnet sind, in den
Bitstrom eingefügt werden.

11. Verfahren nach einem der Ansprüche 5 bis 10, bei dem die
30 Pfade Beschreibungselemente eines XML-Dokuments (XML =
Extensible Markup Language) umfassen.

12. Verfahren nach Anspruch 11, bei dem die Pfade XPATH-Pfade
des XML-Dokuments sind.

35 13. Verfahren nach einem der vorhergehenden Ansprüche, bei
dem die Indexdaten mit einem Codierverfahren binär co-
diert sind, insbesondere mit einem MPEG-Codierverfahren.

14. Verfahren nach Anspruch 13, bei dem das Codierverfahren ein MPEG7-Codierverfahren ist.

5 15. Verfahren zum Codieren einer Datenstruktur, bei dem die Datenelemente der Datenstruktur in einem Indizierungsbaum indiziert sind, wobei ein Bitstrom gemäß einem der vorhergehenden Ansprüche erzeugt wird und der Bitstrom Teil des codierten Datenstroms ist.

10

16. Verfahren zum Decodieren einer Datenstruktur, wobei das Verfahren derart ausgestaltet ist, dass die gemäß Anspruch 12 codierte Datenstruktur decodiert wird.

15

17. Verfahren zum Codieren und Decodieren einer Datenstruktur, umfassend das Verfahren nach Anspruch 15 und das Verfahren nach Anspruch 16.

20

18. Codiervorrichtung, mit der ein Verfahren nach Anspruch 15 durchführbar ist.

19. Decodiervorrichtung mit der ein Verfahren nach Anspruch 16 durchführbar ist.

20. Vorrichtung zum Codieren und Decodieren einer Datenstruktur, mit der ein Verfahren nach Anspruch 17 durchführbar ist.

Zusammenfassung**Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum**

5

Die Erfindung betrifft ein Verfahren zur Erzeugung eines Bitstroms aus einem Indizierungsbaum, wobei der Indizierungsbaum eine Mehrzahl von Hierarchieebenen umfasst und jeder Hierarchieebene ein oder mehrere Indexknoten zugeordnet sind, wobei die Indexknoten Indexdaten enthalten, die in dem Indizierungsbaum nach einem oder mehreren vorgegebenen Kriterien sortiert sind. In dem Verfahren werden die Indexdaten der Indexknoten in den Bitstrom eingefügt und für einen Indexknoten wird jeweils die Information in den Bitstrom eingefügt, an welcher Stelle im Bitstrom sich die Indexdaten von einem oder mehreren Indexknoten der Hierarchieebene, die unterhalb der Hierarchieebene des jeweiligen Knotens liegt, befinden.

Figur 1

20

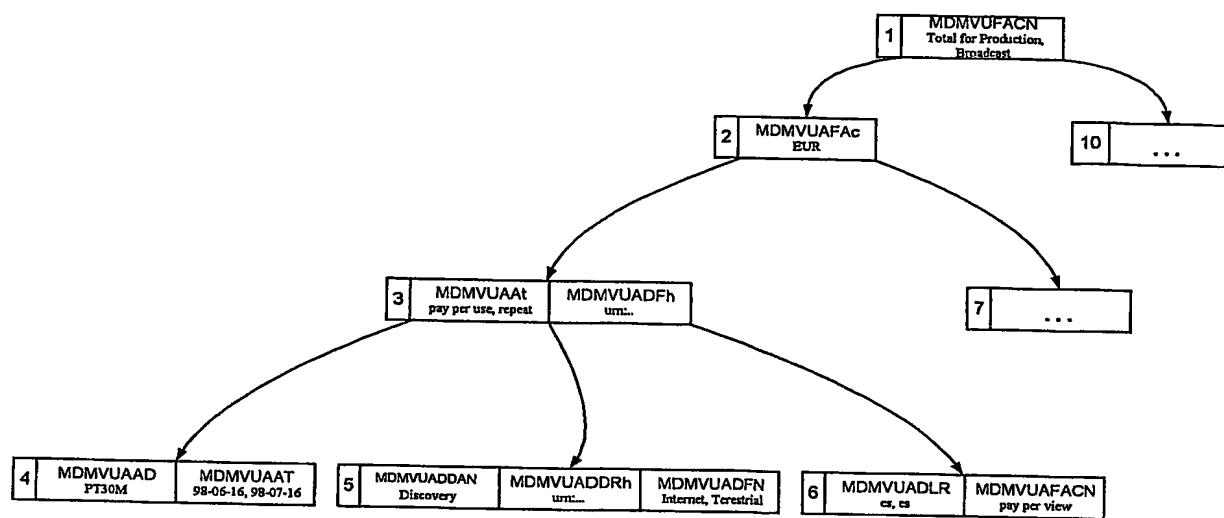


Fig. 1

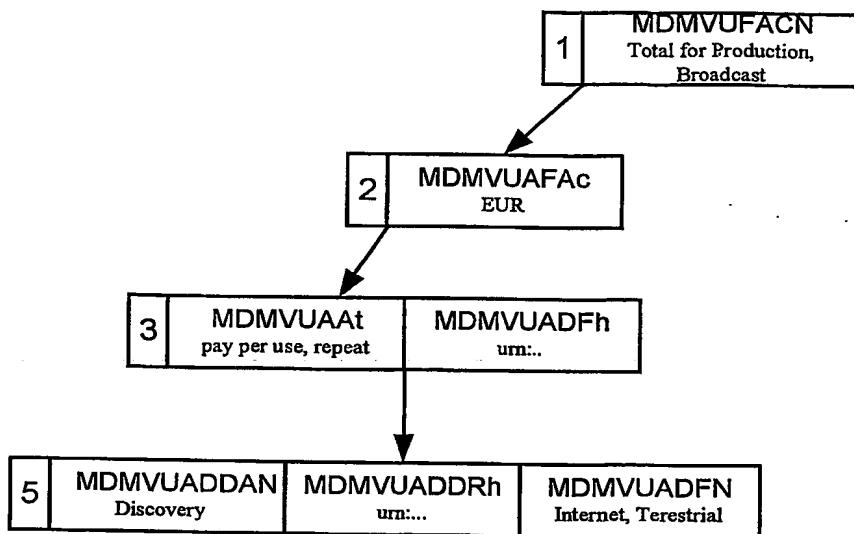


Fig. 2

1	1, MDMVUAFCN, 2, Total for Production, position, Broadcast, position, offset(10)	2	...
3	2, MDRVUAArt, 2, pay per use, position, repeat, position, MDMVUADFh, 1, urn:..., position, offset(5), offset(6)	4	...

Fig. 3

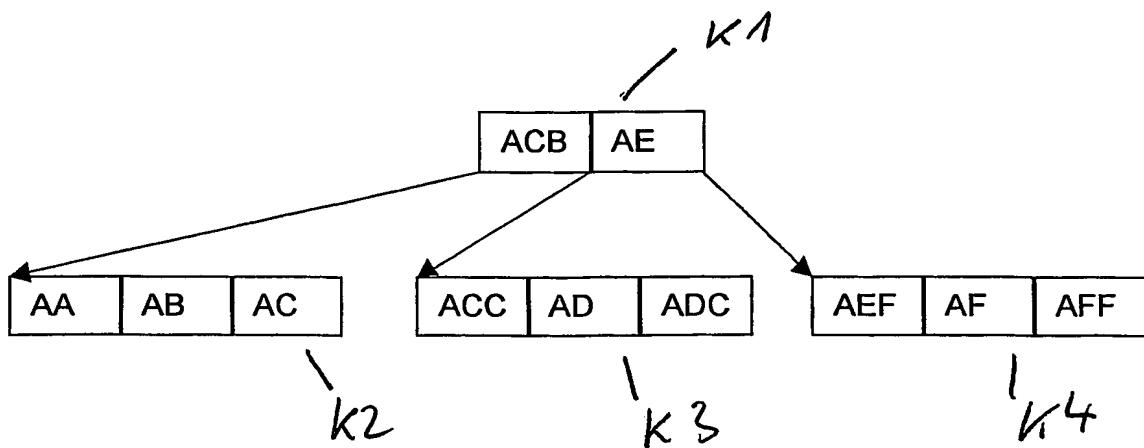


Fig. 4

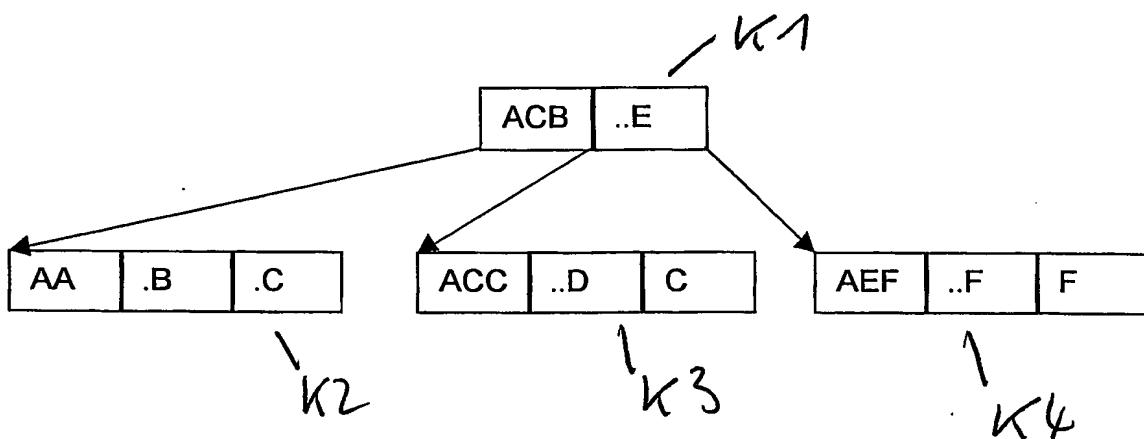


Fig. 5

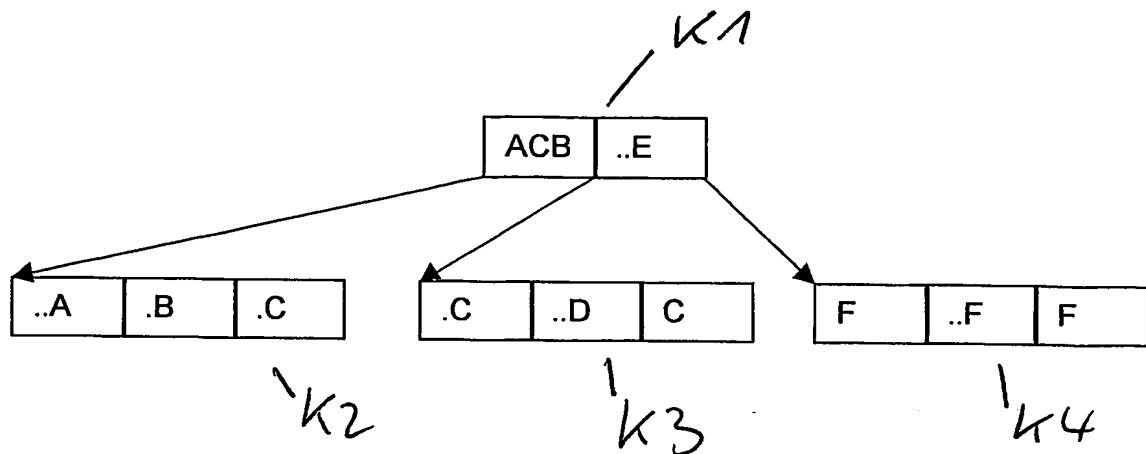


Fig. 6

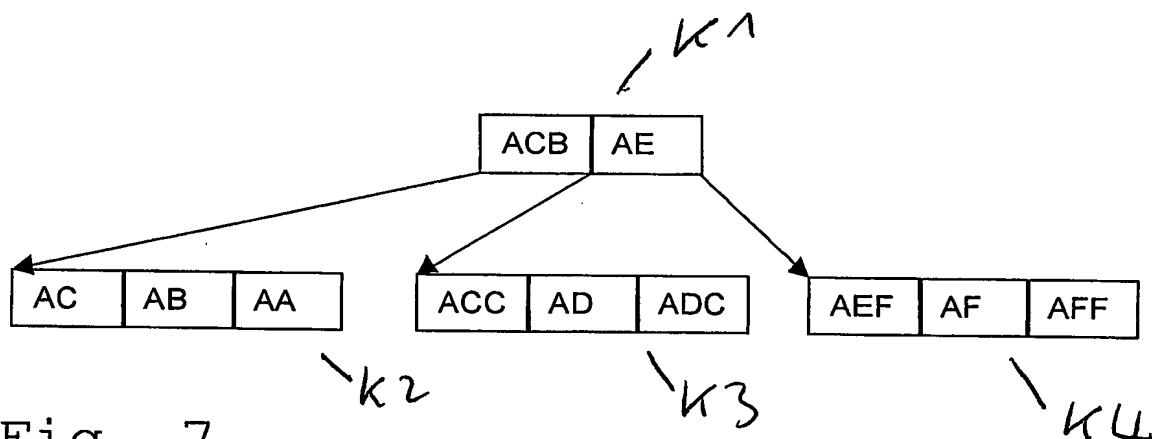


Fig. 7